Geometry Processing (601.458/658)

Misha Kazhdan

"As-Rigid-As-Possible Surface Modeling", Sorkine and Alexa, 2007

Outline

Recall

Quadratic Energies with Affine Constraints

As-Rigid-As-Possible Surface Modeling

Recall: SVD Factorization

Any matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ can be expressed:

$$\mathbf{M} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^\top$$

with:

- $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times n}$ orthogonal (i.e. $\mathbf{U}^\top = \mathbf{U}^{-1}$ and $\mathbf{V}^\top = \mathbf{V}^{-1}$)
- $\mathbf{D} \in \mathbb{R}^{n \times n}$ diagonal (with non-negative entries)

Recall: Procrustes Method

Given two sets of corresponding points $\{\mathbf{p}^1, \dots, \mathbf{p}^n\}$ and $\{\mathbf{q}^1, \dots, \mathbf{q}^n\}$ in $\mathbb{R}^d$, we would like the **translation** minimizing the sum of squared differences:

$$E(\mathbf{t}) = \sum_{i=1}^{n} \left| \left(\mathbf{p}^i + \mathbf{t}\right) - \mathbf{q}^i \right|^2$$

This is the translation taking the center of mas of $\{\mathbf{p}^1, \dots, \mathbf{p}^n\}$ to $\{\mathbf{q}^1, \dots, \mathbf{q}^n\}$:

$$\mathbf{t} = \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{q}^i\right) - \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{p}^i\right)$$

$\Rightarrow$ If each set has its center of mass at the origin, they are optimally aligned.

Recall: Procrustes Method

Given two sets of corresponding points $\{\mathbf{p}^1, \ldots, \mathbf{p}^n\}$ and $\{\mathbf{q}^1, \ldots, \mathbf{q}^n\}$ in $\mathbb{R}^d$, whose centers of mass are at the origin, we would like the **orthogonal transformation** minimizing the sum of squared differences:

$$E(\mathbf{O}) = \sum_{i=1}^{n} \left|\mathbf{O} \cdot \mathbf{p}^i - \mathbf{q}^i\right|^2$$

We can construct the cross-correlation matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$:

$$\mathbf{C}_{ij} = \sum_{k=1}^{n} \mathbf{p}_i^k \cdot \mathbf{q}_j^k$$

Taking the SVD factorization, $\mathbf{C} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^\top$, the minimizing rotation is:

$$\mathbf{O} = \mathbf{V} \cdot \mathbf{U}^\top$$

Recall: Procrustes Method

Given two sets of corresponding points $\{\mathbf{p}^1, \ldots, \mathbf{p}^n\}$ and $\{\mathbf{q}^1, \ldots, \mathbf{q}^n\}$ in $\mathbb{R}^d$, we would like the **translation and orthogonal transformation** minimizing the sum of squared differences:

$$E(\mathbf{t}, \mathbf{O}) = \sum_{i=1}^{n} \left|\left(\mathbf{O} \cdot \mathbf{p}^i + \mathbf{t}\right) - \mathbf{q}^i\right|^2$$

Let $\overline{\mathbf{p}}$ and $\overline{\mathbf{q}}$ be the centers of mass of the two points, and let $\mathbf{O} \in \mathbb{R}^{d\times d}$ be the optimal orthogonal transformation aligning the centered point-sets:

$$\begin{aligned} E(\mathbf{O}) &= \sum_{i=1}^{n} \left|\left(\mathbf{O} \cdot (\mathbf{p}^i - \overline{\mathbf{p}})\right) - \left(\mathbf{q}^i - \overline{\mathbf{q}}\right)\right|^2 \\ &= \sum_{i=1}^{n} \left|\left(\mathbf{O} \cdot \mathbf{p}^i + \underbrace{\boxed{(\overline{\mathbf{q}} - \mathbf{O} \cdot \overline{\mathbf{p}})}}_{\mathbf{t}}\right) - \mathbf{q}^i\right| \end{aligned}$$

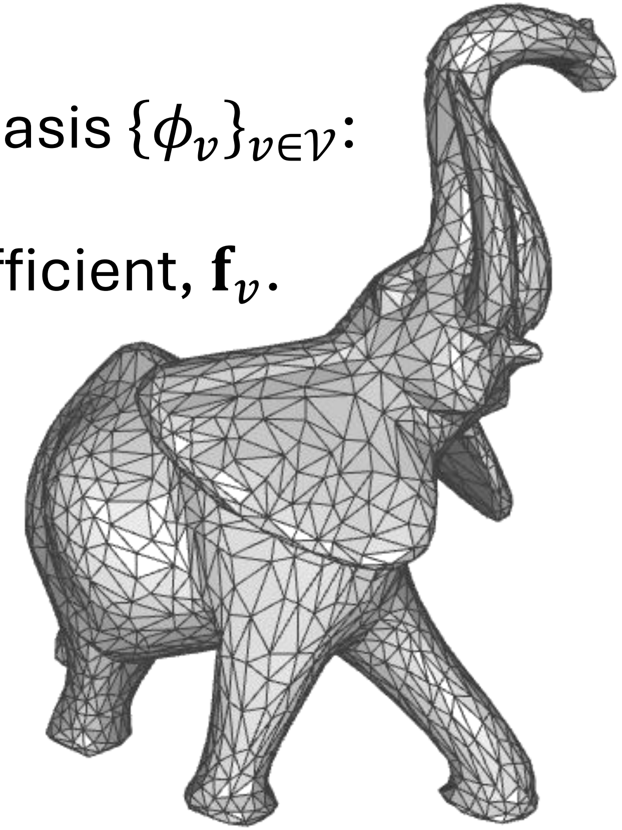
Recall

Given a triangle mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{T}\}$, we denote the space of piecewise linear functions, spanned by the that basis functions as V and the space of piecewise constant cotangent vector fields as $\overline{V}$.

For a scalar function $f \in V$, expressed in terms of the basis $\{\phi_v\}_{v\in\mathcal{V}}$:

$$f = \mathbf{f}_1 \cdot \phi_1 + \cdots + \mathbf{f}_{|\mathcal{V}|} \cdot \phi_{|\mathcal{V}|}$$

the value of the function at vertex $v \in \mathcal{V}$ is the v-th coefficient, $\mathbf{f}_v$.



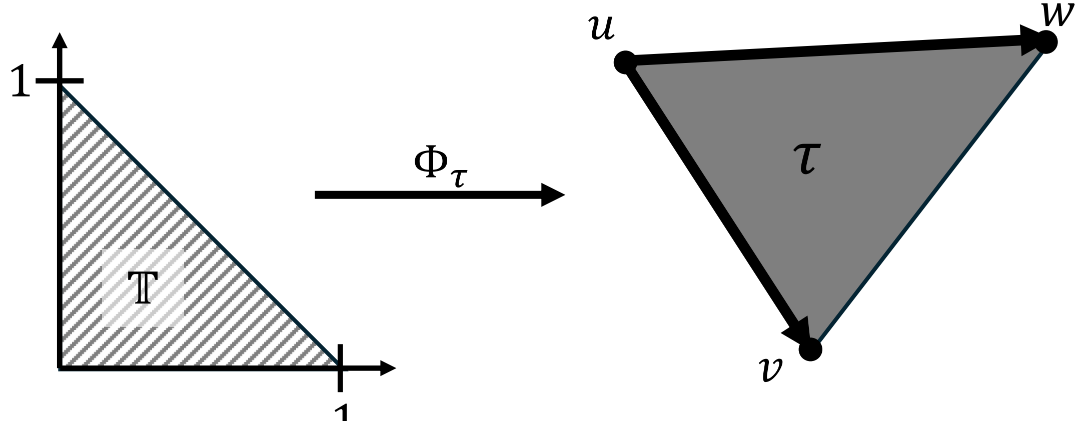
Recall

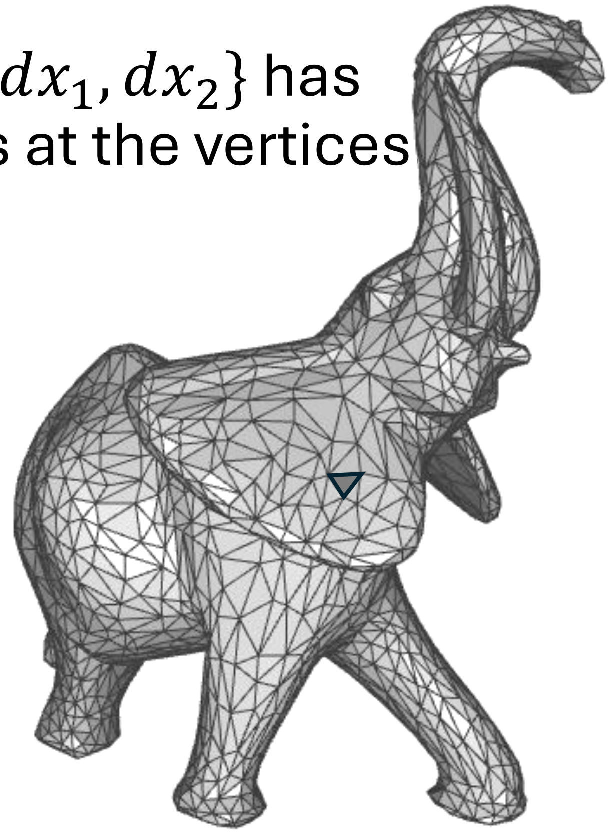
For $\tau = (u, v, w) \in \mathcal{T}$, the associated function on $\mathbb{T}$ is the interpolant:

$$f_\tau(x_1, x_2) = \mathbf{f}_u + (\mathbf{f}_v - \mathbf{f}_u) \cdot x_1 + (\mathbf{f}_w - \mathbf{f}_u) \cdot x_2$$

$\Rightarrow$ The differential of $f_\tau : \mathbb{T} \to \mathbb{R}$, expressed w.r.t. the basis $\{dx_1, dx_2\}$ has coefficients that are the differences between the values at the vertices

$$\mathbf{d}f_\tau = \begin{pmatrix} \mathbf{f}_v - \mathbf{f}_u \\ \mathbf{f}_w - \mathbf{f}_u \end{pmatrix}$$





Recall

Given a target scalar field $h \in V$ and a target cotangent vector field $\zeta \in \overline{V}$, the gradient domain problem seeks the scalar function $f \in V$ minimizing:

$$E(f) = \alpha \cdot \frac{1}{2} \cdot \langle\langle f - h, f - h \rangle\rangle_{\mathcal{M}} + \beta \cdot \frac{1}{2} \cdot \langle\langle df - \zeta, df - \zeta \rangle\rangle_{\mathcal{M}}$$

Expressed w.r.t. the scalar and cotangent vector field bases $\{\phi_v\}_{v \in \mathcal{V}}$ and $\{\eta_\tau^1, \eta_\tau^2\}_{\tau \in \mathcal{T}}$, the coefficients of the solution are given by:

$$\mathbf{f} = (\alpha \cdot \mathbf{M} + \beta \cdot \mathbf{S})^{-1} \cdot \left(\alpha \cdot \mathbf{M} \cdot \mathbf{h} + \beta \cdot \mathbf{D}^\top \cdot \overline{\mathbf{M}} \cdot \boldsymbol{\zeta}\right)$$

with:

- $\mathbf{M}, \mathbf{S} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ the scalar mass and stiffness matrices,
- $\overline{\mathbf{M}} \in \mathbb{R}^{2|\mathcal{T}| \times 2|\mathcal{T}|}$ the mass matrix for cotangent vector fields,
- $\mathbf{D} \in \mathbb{R}^{2|\mathcal{T}| \times |\mathcal{V}|}$ the differential,
- $\mathbf{h} \in \mathbb{R}^{|\mathcal{V}|}$ the target values, and
- $\boldsymbol{\zeta} \in \mathbb{R}^{2|\mathcal{T}||}$ the target differential

Outline

Recall

Quadratic Optimization with Affine Constraints

As-Rigid-As-Possible Surface Modeling

Quadratic Optimization

$$V \xrightarrow{E} \mathbb{R}$$

Recall:

Given a vector space V with basis $\{v_1, \ldots, v_n\}$, a symmetric positive-definite bilinear form $B: V \to V^*$, and a dual vector $l \in V^*$ we can define the energy:

$$E: V \to \mathbb{R}$$
$$v \mapsto \frac{1}{2}[B(v)](v) - l(v)$$

The minimizer is given by:

$$\underset{\{v \in V\}}{\operatorname{argmin}}\, E(v) = B^{-1}(l)$$

$\boxed{E(v) = \frac{1}{2}[B(v)](v) - l(v)}$

Quadratic Optimization

$$\begin{tikzcd} V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^k \\ \mathbb{R} \end{tikzcd}$$

Question:

What happens if we add linear constraints?

That is, we are also given dual vectors $l_1, \ldots, l_k \in V^*$ and seek the minimizer of E subject to the constraints that, for all $1 \leq i \leq c$:

$$l_i(v) = 0$$

Equivalently:

We are given a linear map $C \in \mathrm{Hom}(V, \mathbb{R}^c)$ and seek the minimizer of E subject to the constraints that:

$$C(v) = \mathbf{0}$$

$$\boxed{\begin{array}{c} E(v) = \frac{1}{2}[B(v)](v) - l(v) \\ C(v) = \mathbf{0} \end{array}}$$

$$\begin{array}{ccccc} \mathrm{Ker}(C) & \hookrightarrow & V & \xrightarrow{C} & \mathbb{R}^c \\ & & \downarrow{\scriptstyle E} & & \\ & & \mathbb{R} & & \end{array}$$

Quadratic Optimization

Approach:

Identify the *kernel* of the linear operator C:

$$\mathrm{Ker}(C) = \{v \in V | C(v) = \mathbf{0}\} \subset V$$

This is the subset of vectors in V on which C vanishes.

$\Leftrightarrow$ This is the subset of vectors in V satisfying the constraint.

Since C is linear, for all $v, w \in \mathrm{Ker}(C)$ and all $\alpha, \beta \in \mathbb{R}$:

$$\begin{aligned} C(\alpha \cdot v + \beta \cdot w) &= \alpha \cdot C(v) + \beta \cdot C(w) \\ &= \alpha \cdot \mathbf{0} + \beta \cdot \mathbf{0} \\ &= \mathbf{0} \end{aligned}$$

$\Rightarrow$ The vector $\alpha \cdot v + \beta \cdot w$ is also in the kernel.

$\Rightarrow$ The kernel is a subspace of V.

$$\boxed{\begin{array}{c} E(v) = \frac{1}{2}[B(v)](v) - l(v) \\ C(v) = \mathbf{0} \end{array}}$$

$$\begin{tikzcd}
\mathrm{Ker}(C) \arrow[r, hook, "\iota"] \arrow[dr, "E_C"'] & V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^c \\
& \mathbb{R} &
\end{tikzcd}$$

Quadratic Optimization

Approach:

We would like to optimize the energy over the kernel:

$$\underset{\{u \in \mathrm{Ker}(C)\}}{\operatorname{argmin}} \; E(v)$$

Let $\iota \in \mathrm{Hom}(\mathrm{Ker}(C), V)$ be the *injection* operator:

$$\begin{aligned} \iota : \mathrm{Ker}(C) &\to V \\ u &\mapsto u \end{aligned}$$

We can define the energy on $\mathrm{Ker}(C)$ by restricting the energy from V:

$$\begin{aligned} E_C(u) &= E\big(\iota(u)\big) \\ &= \tfrac{1}{2}[B(\iota(u))](\iota(u)) - l\big(\iota(u)\big) \\ &= \tfrac{1}{2}[\iota^* \circ B \circ \iota](v) - [\iota^* \circ l](u) \end{aligned}$$

$$\boxed{\begin{array}{c} E(v) = \frac{1}{2}[B(v)](v) - l(v) \\ C(v) = \mathbf{0} \end{array}}$$

$$\begin{tikzcd} \mathrm{Ker}(C) \arrow[r, hook, "\iota"] \arrow[dr, "E_C"'] & V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^c \\ & \mathbb{R} & \end{tikzcd}$$

Quadratic Optimization

Approach:

$$E_C(u) = \frac{1}{2}[\iota^* \circ B \circ \iota](u) - [\iota^* \circ l](u)$$

$\Rightarrow$ The minimizer of the energy is the vector $u \in \mathrm{Ker}(C)$ satisfying:

$$\underset{\{u \in \mathrm{Ker}(C)\}}{\operatorname{argmin}} \; E_C(u) = (\iota^* \circ B \circ \iota)^{-1}\big(\iota^*(l)\big)$$

Or, as a vector in V, the minimizer is:

$$\underset{\{v \in V | C(v) = \mathbf{0}\}}{\operatorname{argmin}} \; E(v) = \iota\Big((\iota^* \circ B \circ \iota)^{-1}\big(\iota^*(l)\big)\Big)$$

$$E(v) = \frac{1}{2}[B(v)](v) - l(v)$$
$$C(v - \hat{v}) = \mathbf{0}$$

$$\begin{tikzcd} \mathrm{Ker}(C) \arrow[r, hook, "\iota"] \arrow[dr, "E_C"'] & V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^c \\ & \mathbb{R} & \end{tikzcd}$$

Quadratic Optimization

Question:

What happens if the constraints are **affine** instead of **linear**?

That is, we are also given a vector $\hat{v} \in V$ and seek the minimizer of E subject to the constraints that, for all $1 \leq i \leq c$:

$$l_i(v - \hat{v}) = 0$$
$$\Updownarrow$$
$$C(v - \hat{v}) = \mathbf{0}$$

$$\boxed{\begin{gathered} E(v) = \tfrac{1}{2}[B(v)](v) - l(v) \\ C(v - \hat{v}) = \mathbf{0} \end{gathered}}$$

$$\begin{tikzcd} \mathrm{Ker}(C) \arrow[r, hook, "\iota"] \arrow[dr, "E_C"'] & V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^c \\ & \mathbb{R} & \end{tikzcd}$$

Quadratic Optimization

Observation:

The **affine** space:

$$\mathrm{Ker}(C) + \hat{v}$$

consists of vectors that satisfy the constraints.

$\Rightarrow$ Minimizing the energy E over the subset of vectors satisfying $C(v - \hat{v}) = \mathbf{0}$ is equivalent to minimizing the energy:

$$E_C(u) = E(\iota(u) + \hat{v})$$

over $u \in \mathrm{Ker}(C)$ and then adding $\hat{v}$ to the minimizer.

$$\underset{\{v \in V | C(v - \hat{v}) = \mathbf{0}\}}{\mathrm{argmin}} E(v) = \iota\left(\underset{\{u \in \mathrm{Ker}(C)\}}{\mathrm{argmin}} E_C(u)\right) + \hat{v}$$

$$\boxed{\begin{array}{c} E(v) = \frac{1}{2}[B(v)](v) - l(v) \\ C(v-\hat{v}) = \mathbf{0} \end{array}}$$

$$\begin{tikzcd}
\mathrm{Ker}(C) \arrow[r, hook, "\iota"] \arrow[dr, "E_C"'] & V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^c \\
& \mathbb{R} &
\end{tikzcd}$$

Quadratic Optimization

Expanding

$$\begin{aligned}
E_C(u) &= E(\iota(u) + \hat{v}) \\
&= \tfrac{1}{2}[B(\iota(u)+\hat{v})](\iota(u)+\hat{v}) - l(\iota(u)+\hat{v}) \\
&= \tfrac{1}{2}\Big([B(\iota(u))](\iota(u)) + [B(\hat{v})](\hat{v}) + [B(\hat{v})](\iota(u)) + [B(\iota(u))](\hat{v})\Big) \\
&\qquad\qquad -l(\iota(u)) - l(\hat{v}) \\
&= \tfrac{1}{2}\Big([(\iota^* \circ B \circ \iota)(u)](u) + [B(\hat{v})](\hat{v}) + 2[B(\hat{v})](\iota(u))\Big) - l(\iota(u)) - l(\hat{v}) \\
&= \tfrac{1}{2}[(\iota^* \circ B \circ \iota)(u)](u) + [B(\hat{v})](\iota(u)) - l(\iota(u)) - l(\hat{v}) + \tfrac{1}{2}[B(\hat{v})](\hat{v}) \\
&= \tfrac{1}{2}[(\iota^* \circ B \circ \iota)(u)](u) + [(\iota^* \circ B)(\hat{v})](u) - [\iota^*(l)](u) - l(\hat{v}) + \tfrac{1}{2}[B(\hat{v})](\hat{v}) \\
&= \tfrac{1}{2}[(\iota^* \circ B \circ \iota)(u)](u) - [\iota^*(l) - (\iota^* \circ B)(\hat{v})](u) \cancel{- l(\hat{v}) + \tfrac{1}{2}[B(\hat{v})](\hat{v})}
\end{aligned}$$

Taking the differential and setting to zero:

$$\underset{\{u \in \mathrm{Ker}(C)\}}{\operatorname{argmin}}\; E_C(u) = (\iota^* \circ B \circ \iota)^{-1}\Big(\iota^*\big(l - B(\hat{v})\big)\Big)$$

$$\boxed{\begin{gathered} E(v) = \tfrac{1}{2}[B(v)](v) - l(v) \\ C(v - \hat{v}) = \mathbf{0} \end{gathered}}$$

Quadratic Optimization

$$\begin{tikzcd} \mathrm{Ker}(C) \arrow[r, hook, "\iota"] \arrow[dr, "E_C"'] & V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^c \\ & \mathbb{R} & \end{tikzcd}$$

$$\begin{aligned} \underset{\{v \in V \mid C(u - \hat{v}) = \mathbf{0}\}}{\operatorname{argmin}} E(v) &= \iota\left(\underset{\{u \in \mathrm{Ker}(C)\}}{\operatorname{argmin}} E(\iota(v) + \hat{v}) \right) + \hat{v} \\ &= \iota\left((\iota^* \circ B \circ \iota)^{-1} \left(\iota^*\left(l - B(\hat{v})\right)\right)\right) + \hat{v} \end{aligned}$$

$\Rightarrow$ If we have a basis $\{v_1, \ldots, v_n\}$ for V and a basis $\{w_1, \ldots, w_m\}$ for $\mathrm{Ker}(C)$ we can express our operators/vectors in terms of the basis:

- $B \to \mathbf{B} \in \mathbb{R}^{n \times n}$
- $\iota \to \boldsymbol{\iota} \in \mathbb{R}^{n \times m}$
- $l \to \mathbf{l} \in \mathbb{R}^{n}$
- $\hat{v} \to \hat{\mathbf{v}} \in \mathbb{R}^{n}$

$\Rightarrow$ With respect to these bases, the solution would be:

$$\mathbf{v} = \boldsymbol{\iota} \cdot (\boldsymbol{\iota}^\top \cdot \mathbf{B} \cdot \boldsymbol{\iota})^{-1} \cdot \boldsymbol{\iota}^\top \cdot (\mathbf{l} - \mathbf{B} \cdot \hat{\mathbf{v}}) + \hat{\mathbf{v}}$$

$$E(v) = \frac{1}{2}[B(v)](v) - l(v)$$
$$C(v - \hat{v}) = \mathbf{0}$$

Quadratic Optimization

$$\begin{tikzcd}
\mathrm{Ker}(C) \arrow[r, hook, "\iota"] \arrow[dr, "E_C"'] & V \arrow[r, "C"] \arrow[d, "E"] & \mathbb{R}^c \\
& \mathbb{R} &
\end{tikzcd}$$

$$\begin{aligned}
\underset{\{v \in V | C(u - \hat{v}) = \mathbf{0}\}}{\operatorname{argmin}} E(v) &= \iota\left(\underset{\{u \in \mathrm{Ker}(C)\}}{\operatorname{argmin}} E(\iota(v) + \hat{v})\right) + \hat{v} \\
&= \iota\left((\iota^* \circ B \circ \iota)^{-1}\left(\iota^*\left(l - B(\hat{v})\right)\right)\right) + \hat{v}
\end{aligned}$$

$\Rightarrow$ If we have a basis $\{v_1, \dots, v_n\}$ for V and a basis $\{w_1, \dots, w_m\}$ for $\mathrm{Ker}(C)$ we can express our operators/vectors in terms of the basis:

- $B \to \mathbf{B} \in \mathbb{R}^{n \times n}$
- ι
- $l \to \mathbf{l}$
- $\hat{v} \to \hat{\mathbf{v}}$

In general, it is not easy to compute the kernel/injection!

There is a class of problems for which this is easy.

$\Rightarrow$ With respect to these bases, the solution would be:

$$\mathbf{v} = \boldsymbol{\iota} \cdot (\boldsymbol{\iota}^\top \cdot \mathbf{B} \cdot \boldsymbol{\iota})^{-1} \cdot \boldsymbol{\iota}^\top \cdot (\mathbf{l} - \mathbf{B} \cdot \hat{\mathbf{v}}) + \hat{\mathbf{v}}$$

Quadratic Optimization

$\mathbf{1}\ \mathbf{2}\ 3\ 4\ 5\ \mathbf{6}\ 7\ 8\ \mathbf{9} \cdots n$

$\boldsymbol{\mathcal{C}}$

Simple Kernel:

Given a basis $\{v_1, \ldots, v_n\}$, supposes we want to minimize:

$$E(v) = \frac{1}{2}[B(v)](v) - l(v)$$

locking a subset of the coefficients to prescribed values.

Formally:

For a subset $\mathcal{C} \subset \{1, \ldots, n\}$ and for target coefficients $\{\alpha_c\}_{c\in\mathcal{C}} \subset \mathbb{R}$, we want to minimize the energy E subject to the constraints:

$$v_c^*(v) = \alpha_c \qquad \forall c \in \mathcal{C}$$

$$\Leftrightarrow v_c^*(v - \alpha_c \cdot v_c^*) = 0 \qquad \forall c \in \mathcal{C}$$

Setting, $\hat{v} = \sum_{c\in\mathcal{C}} \alpha_c \cdot v_c$, this is equivalent to the constraint:

$$v_c^*(v - \hat{v}) = 0 \qquad \forall c \in \mathcal{C}$$

Quadratic Optimization

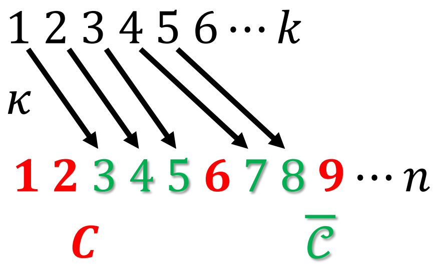
$\mathbf{1}\ \mathbf{2}\ 3\ 4\ 5\ \mathbf{6}\ 7\ 8\ \mathbf{9} \cdots n$

$\boldsymbol{C}$

Kernel:

The kernel is the span of the basis vectors not in $\mathcal{C}$ – the vectors with coefficients that are **not** constrained.

Quadratic Optimization



Kernel:

The kernel is the span of the basis vectors not in $\mathcal{C}$ – the vectors with coefficients that are **not** constrained.

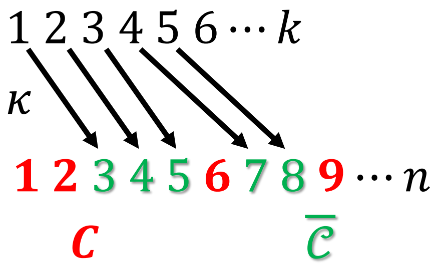
Setting $\overline{\mathcal{C}} = \{1, \ldots, n\} \setminus \mathcal{C}$, the kernel is:

$$\mathrm{Ker}(C) = \mathrm{Span}\left(\{v_{\bar{c}}\}_{\bar{c} \in \overline{\mathcal{C}}}\right)$$

and has dimension $k = n - |\mathcal{C}|$.

Let $\kappa\colon \{1, \ldots, k\} \to \overline{\mathcal{C}}$ be the map taking a values less than or equal to the kernel dimension and returning the associated index in $\overline{\mathcal{C}}$.

Quadratic Optimization



Kernel:

We can define a basis for the kernel:

$$\{w_1 = v_{\kappa(1)}, \dots, w_k = v_{\kappa(k)}\}$$

With respect to the bases $\{v_1, \dots, v_n\}$ and $\{w_1, \dots, w_k\}$ the injection is the matrix $\boldsymbol{\iota} \in \mathbb{R}^{n\times k}$ with:

$$\boldsymbol{\iota}_{ij} = \begin{cases} 1 & i = \kappa(j) \\ 0 & \text{otherwise} \end{cases}$$

We also have $\hat{v} = \sum_{c\in\mathcal{C}} \alpha_c \cdot v_c$, which w.r.t. the basis $\{v_1, \dots, v_n\}$ is represented by the vector $\hat{\mathbf{v}} \in \mathbb{R}^n$:

$$\hat{\mathbf{v}}_i = \begin{cases} \alpha_i & i \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases}$$

Quadratic Optimization

⇒ The minimizer satisfying the interpolation constraints is:

$$\mathbf{v} = \underbrace{\underbrace{\underbrace{\underbrace{\boldsymbol{\iota}}_{\in\mathbb{R}^{n\times k}} \cdot (\underbrace{\boldsymbol{\iota}^\top \cdot \mathbf{B} \cdot \boldsymbol{\iota}}_{\in\mathbb{R}^{k\times k}})^{-1}}_{\in\mathbb{R}^{n\times k}} \cdot \underbrace{\underbrace{\boldsymbol{\iota}^\top}_{\in\mathbb{R}^{k\times n}}}_{\in\mathbb{R}^{k\times n}} \cdot \underbrace{(\underbrace{\mathbf{l}}_{\in\mathbb{R}^{n}} - \underbrace{\mathbf{B} \cdot \hat{\mathbf{v}}}_{\in\mathbb{R}^{n}})}_{\in\mathbb{R}^{n}}}_{\in\mathbb{R}^{n}} + \underbrace{\underbrace{\underbrace{\hat{\mathbf{v}}}_{\in\mathbb{R}^{n}}}_{\in\mathbb{R}^{n}}}_{\in\mathbb{R}^{n}}}_{\in\mathbb{R}^{n}}$$

Sanity Check (Dimensions):

- $\mathbf{B} \in \mathbb{R}^{n\times n}$
- $\boldsymbol{\iota} \in \mathbb{R}^{n\times k}$
- $\boldsymbol{\iota}^\top \in \mathbb{R}^{k\times n}$
- $\mathbf{l} \in \mathbb{R}^{n}$
- $\hat{\mathbf{v}} \in \mathbb{R}^{n}$

Quadratic Optimization

⇒ The minimizer satisfying the interpolation constraints is:

$$\mathbf{v} = \boldsymbol{\iota} \cdot (\boldsymbol{\iota}^\top \cdot \mathbf{B} \cdot \boldsymbol{\iota})^{-1} \cdot \left(\boldsymbol{\iota}^\top \cdot (\mathbf{l} - \mathbf{B} \cdot \hat{\mathbf{v}})\right) + \hat{\mathbf{v}}$$

$$v = \underbrace{\underbrace{\underbrace{\underbrace{\iota}_{\in \mathrm{Hom}(K,V)}}_{\in \mathrm{Hom}(K,V)} \Big(\underbrace{(\underbrace{\iota^* \circ B \circ \iota}_{\in \mathrm{Hom}(K,K^*)})^{-1}}_{\in \mathrm{Hom}(K^*,K)}}_{\in \mathrm{Hom}(K^*,V)} \circ \underbrace{\underbrace{\underbrace{\iota^*}_{\in \mathrm{Hom}(V^*,K^*)}(\underbrace{l - B \circ \hat{v}}_{\in V^*})}_{\in K^*}}_{\in K^*}\Big) + \underbrace{\underbrace{\underbrace{\hat{v}}_{\in V}}_{\in V}}_{\in V}}_{\in V}$$

Sanity Check (Spaces):

- $B \in \mathrm{Hom}(V, V^*)$
- $\iota \in \mathrm{Hom}(K, V)$
- $\iota^* \in \mathrm{Hom}(V^*, K^*)$
- $l \in V^*$
- $\hat{v} \in V$

$\boxed{\text{Denoting } K \equiv \mathrm{Ker}(C)}$

Outline

Recall

Quadratic Energies with Linear Constraints

As-Rigid-As-Possible Surface Modeling

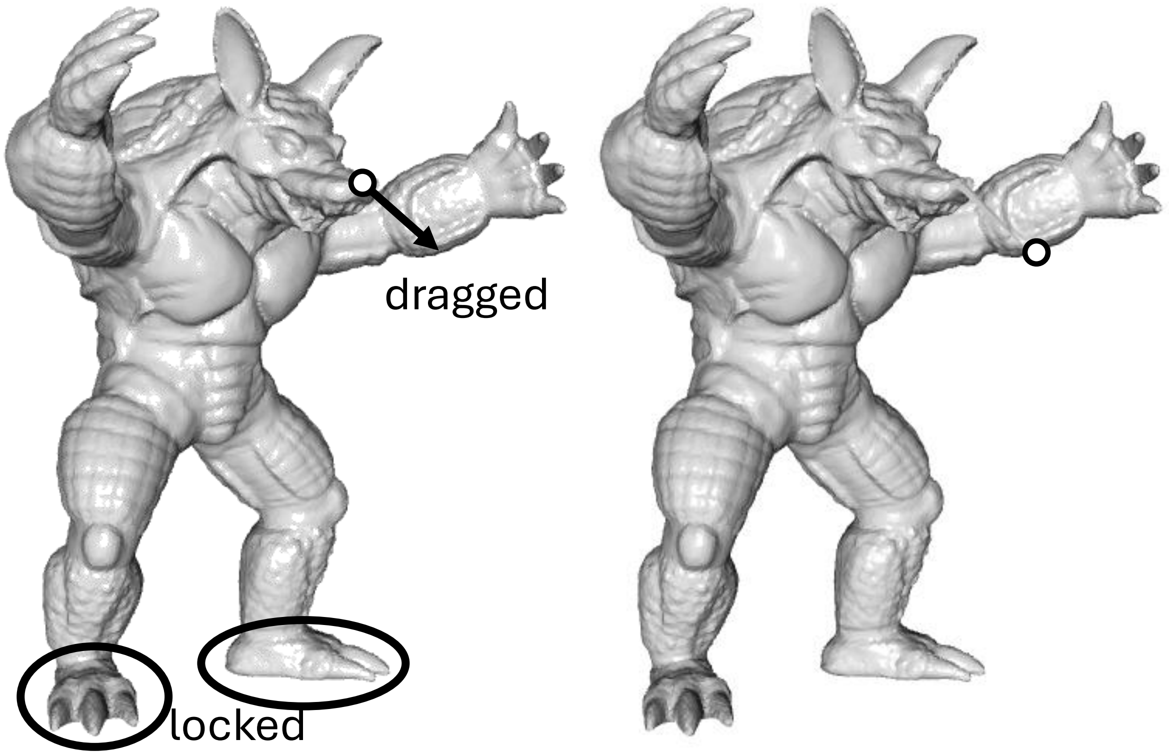
As-Rigid-As-Possible Deformation

Goal:

Given a mesh we would like to support select-and-drag deformation.

Would like to describe a pose by selecting a few vertices and prescribing their target positions.

Just fixing/moving the prescribed vertices creates a “distorted” mesh.



As-Rigid-As-Possible Deformation

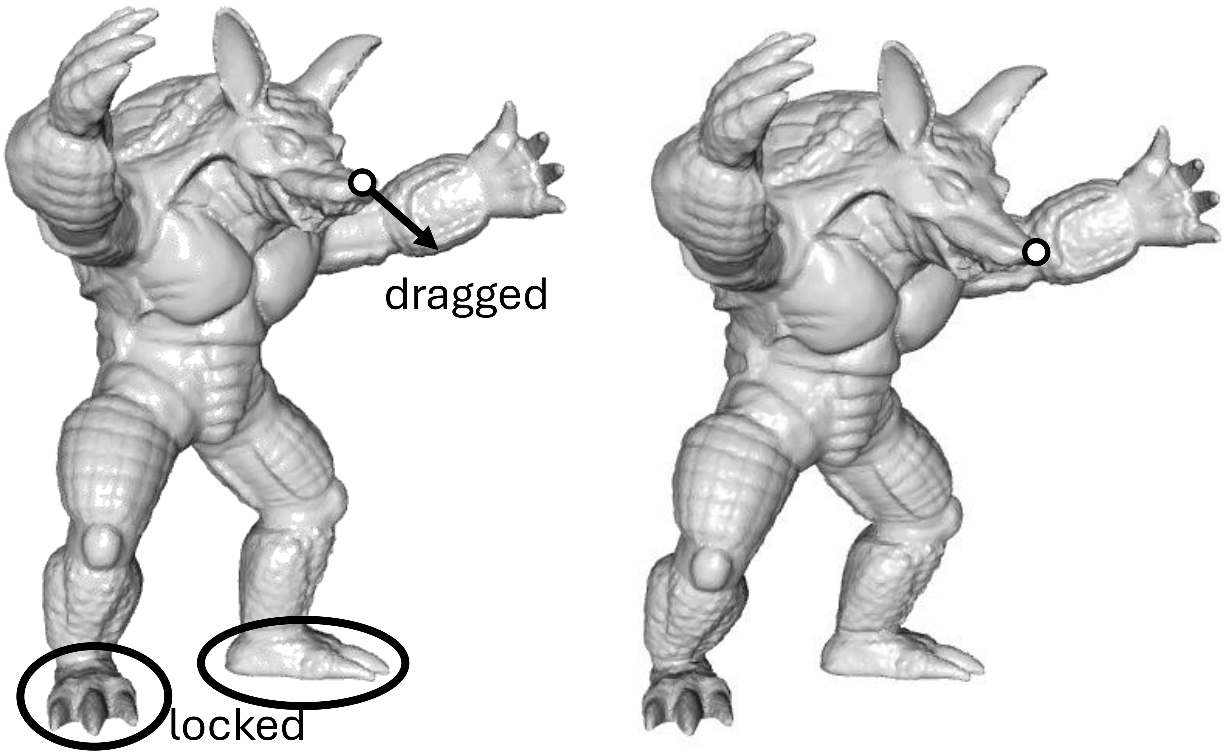
Goal:

Given a mesh we would like to support select-and-drag deformation.

Would like to describe a pose by selecting a few vertices and prescribing their target positions.

Just fixing/moving the prescribed vertices creates a “distorted” mesh.

Want the rest of the surface to move with the constrained vertices.



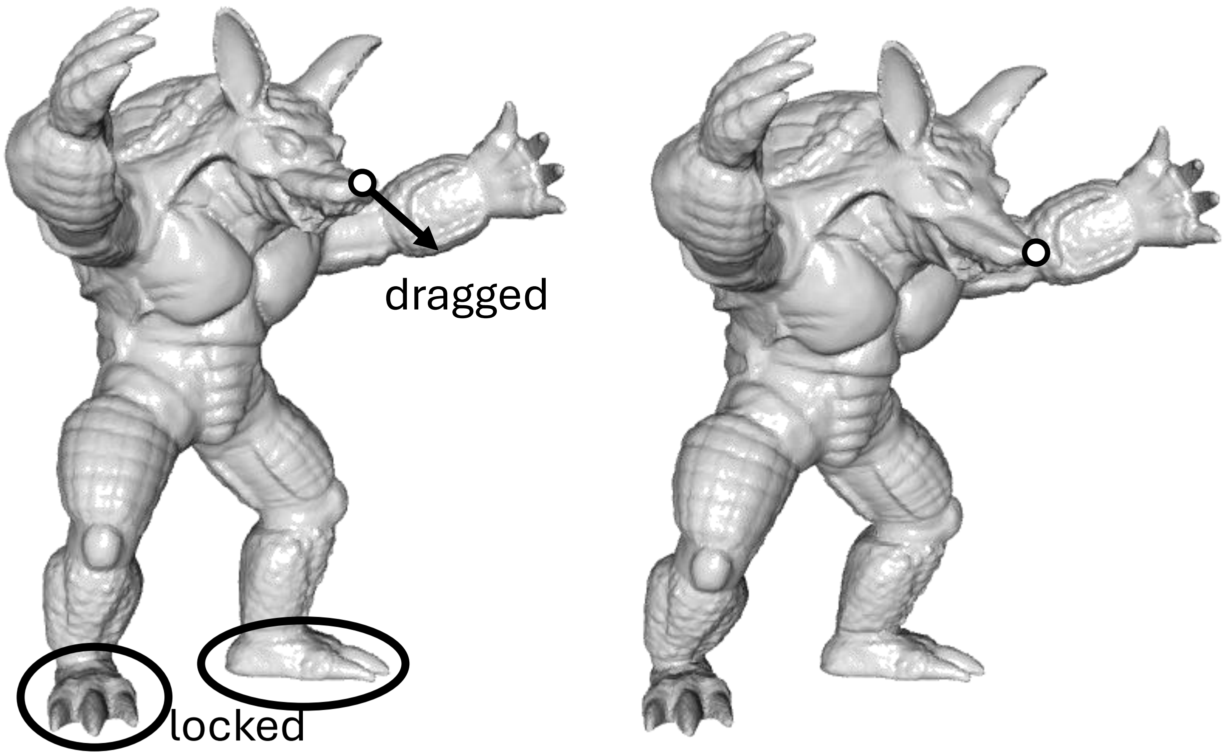
As-Rigid-As-Possible Deformation

Question:

When is the deformed mesh not “distorted”?

Answer:

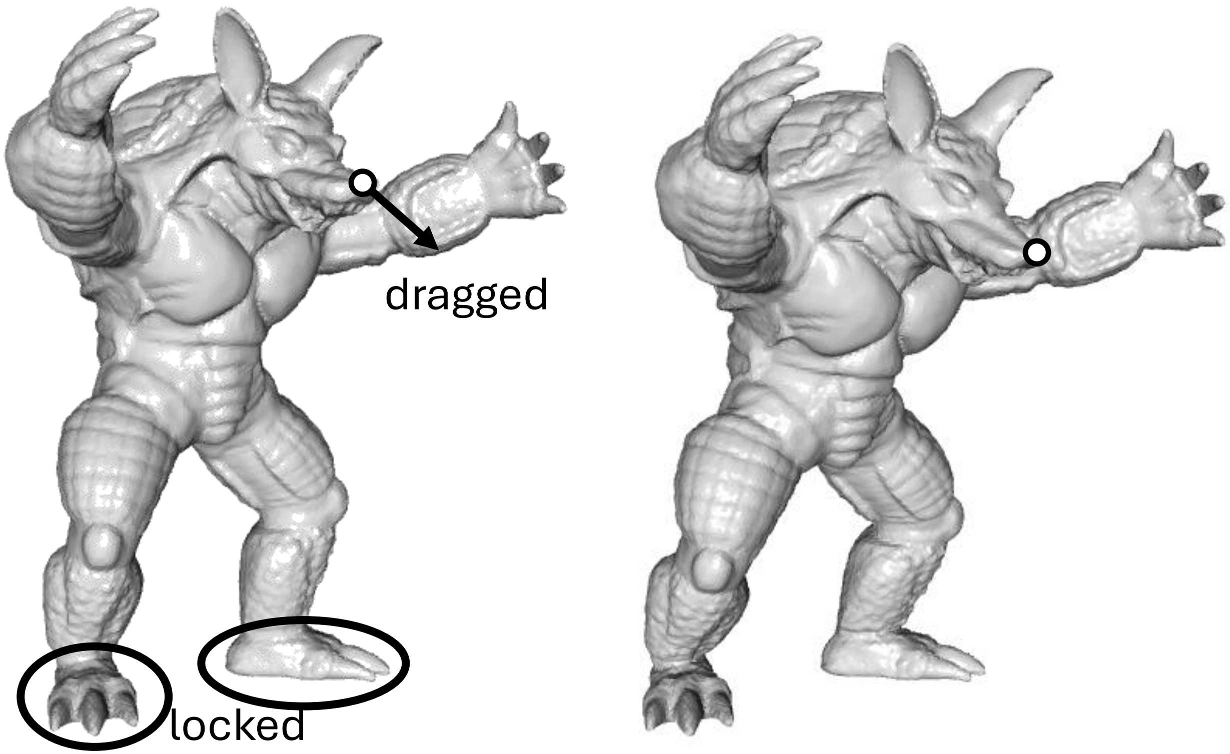
When the source triangles are transformed by an **isometry**.



As-Rigid-As-Possible Deformation

Approach:

1. Set the constrained vertices to their target positions.
2. [Local] Compute target *isometric* deformation per triangle
3. [Global] Fit vertex positions to the target deformation
4. Since the target won't be met exactly, go back to step 2.



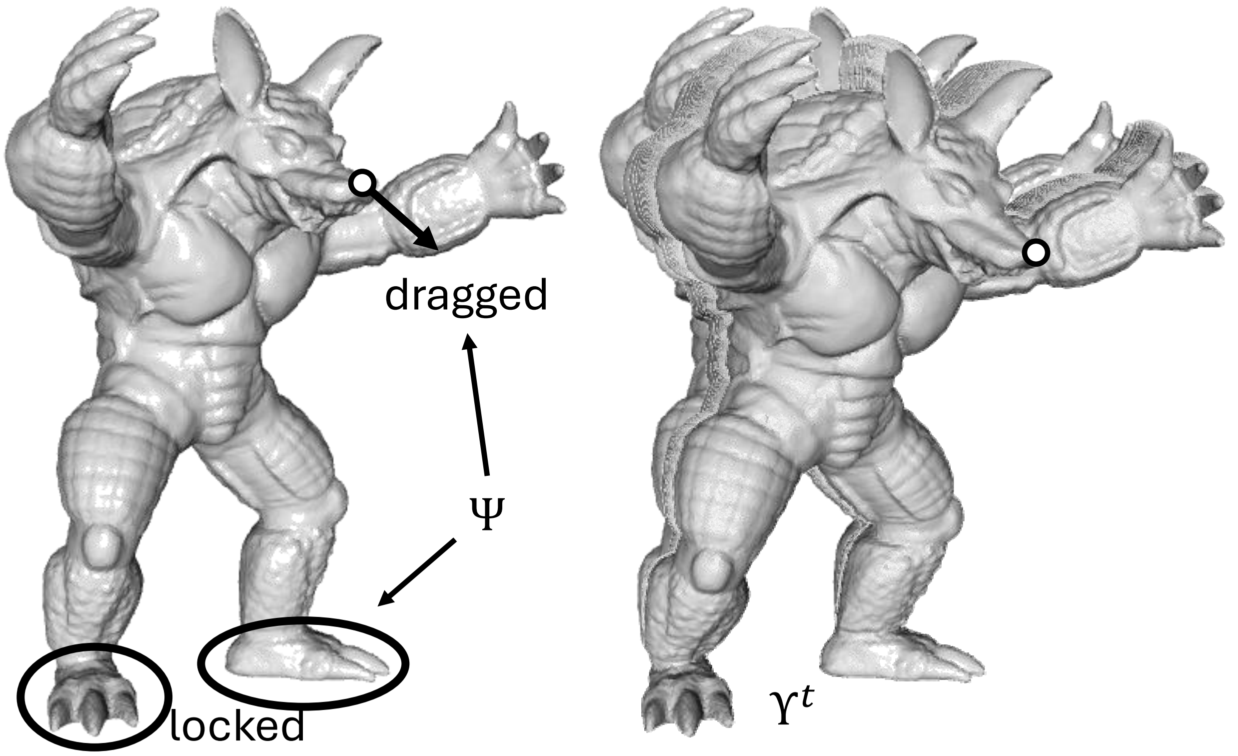
As-Rigid-As-Possible Deformation

Notation:

Denote the set of constrained vertices $\mathcal{C} \subset \mathcal{V}$.

Let $\Psi: \mathcal{C} \to \mathbb{R}^3$ be the assignment of positions to the constrained vertices.

Let $\Upsilon^t: \mathcal{V} \to \mathbb{R}^3$ be the assignment of positions to the mesh vertices at iteration t.
($\Upsilon^t(c) = \Psi(c)$ for all $c \in \mathcal{C}$ and all t).



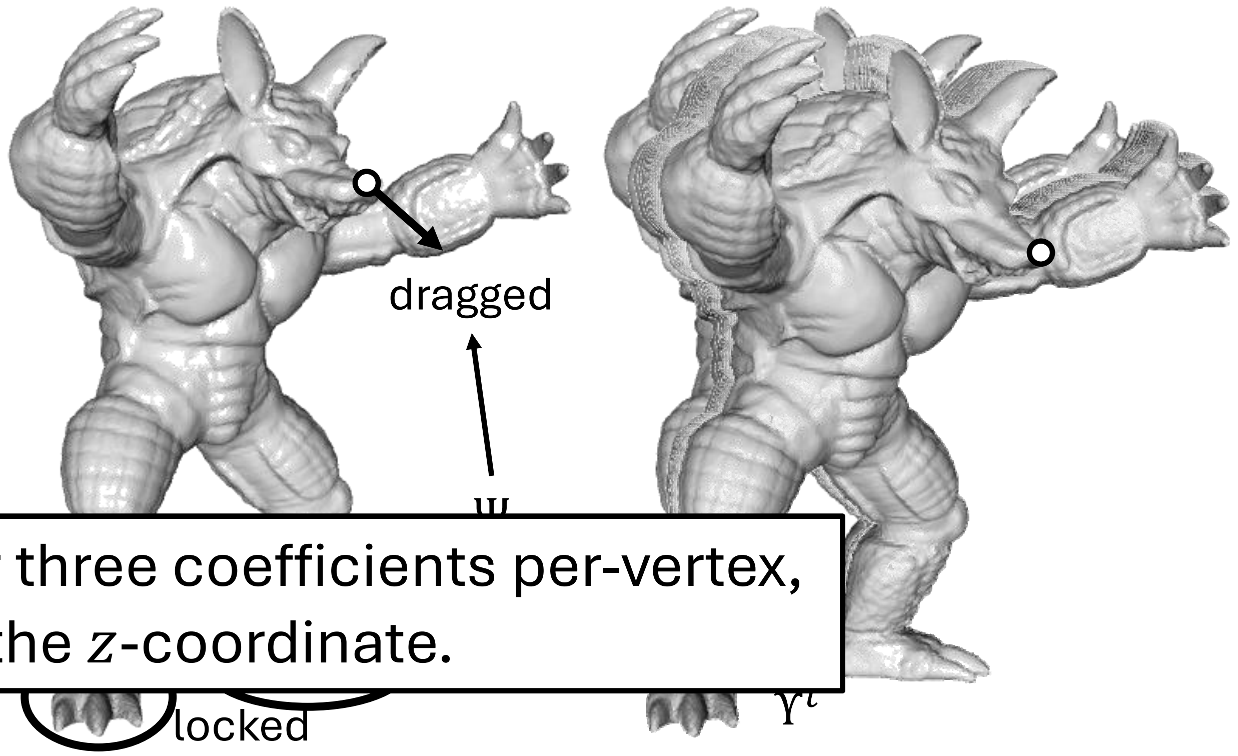
As-Rigid-As-Possible Deformation

Notation:

Denote the set of constrained vertices $\mathcal{C} \subset \mathcal{V}$.

Let $\Psi: \mathcal{C} \to \mathbb{R}^3$ be the assignment of positions to the constrained vertices.

Let $\Upsilon^t: \mathcal{V} \to \mathbb{R}^3$ be the assignment of positions to the mesh vertices at iteration t.
($\Upsilon^t(c) = \Psi(c)$ for all $c \in \mathcal{C}$ and all t).



Note that $\Upsilon^t: \mathcal{V} \to \mathbb{R}^3$, so we're looking for three coefficients per-vertex, one for the x-, one for the y-, and one for the z-coordinate.

As-Rigid-As-Possible Deformation

Local:

At time-step t, we would like to compute the cotangent vector field ζ^{t+1} describing the target differential of the deformation Υ^{t+1} at time-step $t+1$.

We do this greedily by considering each triangle in $\mathcal{T}$ independently.

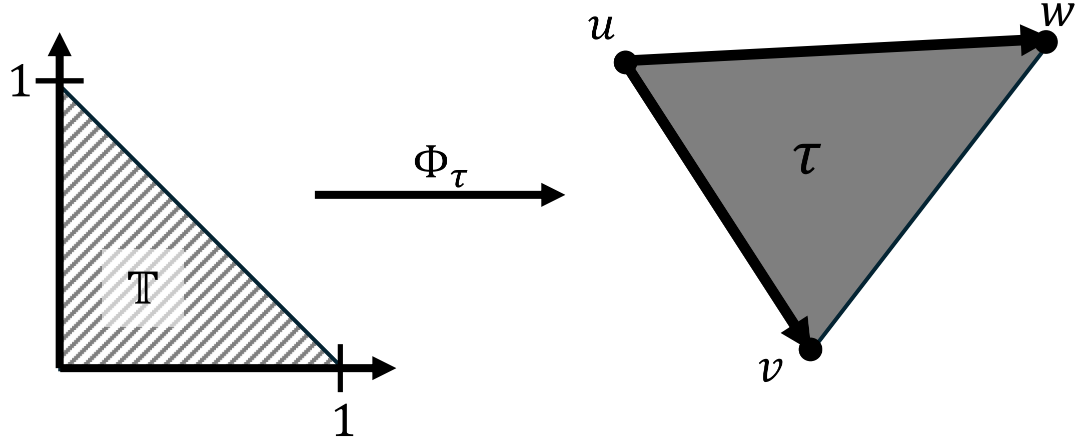
As-Rigid-As-Possible Deformation

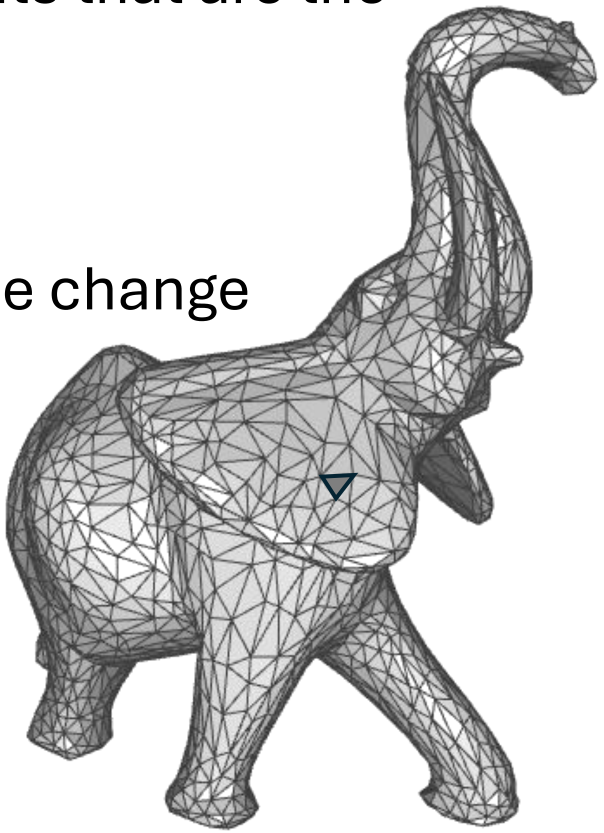
Recall:

For a scalar function $f \in V$ and a triangle $\tau = (u, v, w) \in \mathcal{T}$, the differential of f_τ, expressed w.r.t. the basis $\{dx_1, dx_2\}$ has coefficients that are the differences between the values of f at the vertices:

$$\mathbf{d}f_\tau = \begin{pmatrix} \mathbf{f}_v - \mathbf{f}_u \\ \mathbf{f}_w - \mathbf{f}_u \end{pmatrix}$$

$\Rightarrow$ Prescribing the differential is the same as prescribing the change in the function's values from u to v and from u to w.





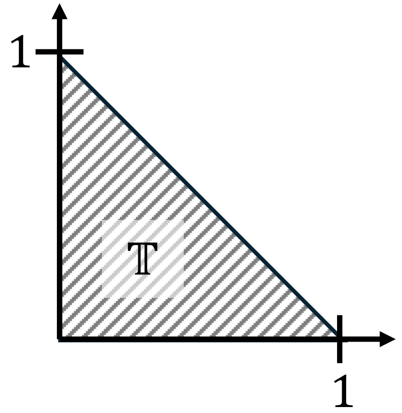
As-Rigid-As-Possible Deformation

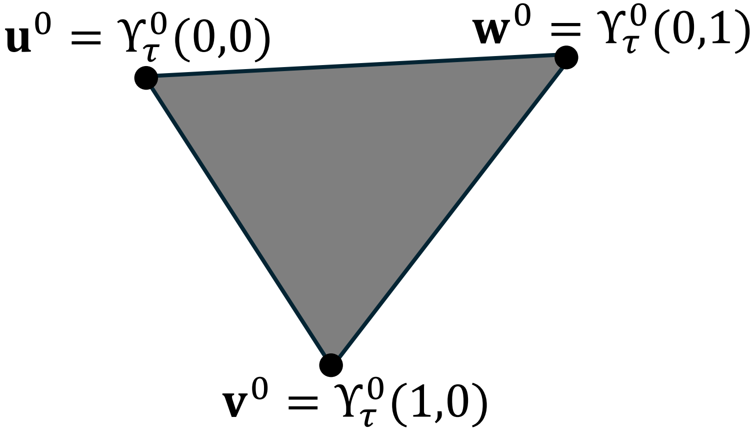
Local:

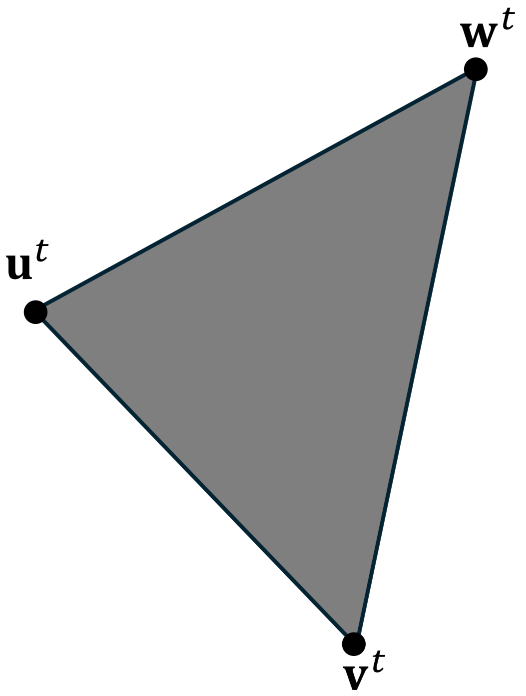
Given a triangle $\tau = (u, v, w) \in \mathcal{T}$, it has assigned positions from the initial embedding, $\Upsilon^0 \colon \mathcal{V} \to \mathbb{R}^3$ and from the current embedding $\Upsilon^t \colon \mathcal{V} \to \mathbb{R}^3$.

We denote the vertex positions:

$$\mathbf{u}^0 \equiv \Upsilon^0(u),\ \mathbf{v}^0 \equiv \Upsilon^0(v),\ \mathbf{w}^0 \equiv \Upsilon^0(w)$$
$$\mathbf{u}^t \equiv \Upsilon^t(u),\ \mathbf{v}^t \equiv \Upsilon^t(v), \mathbf{w}^t \equiv \Upsilon^t(w)$$





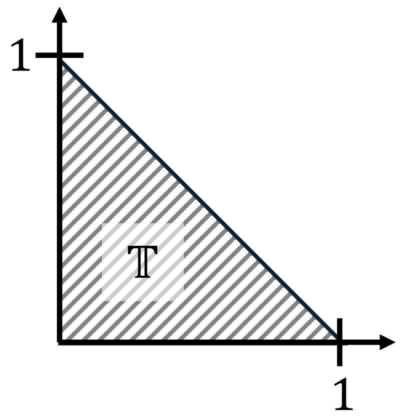


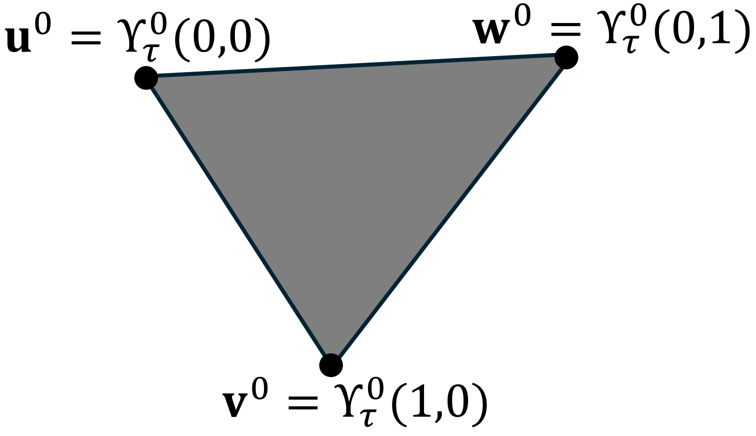
As-Rigid-As-Possible Deformation

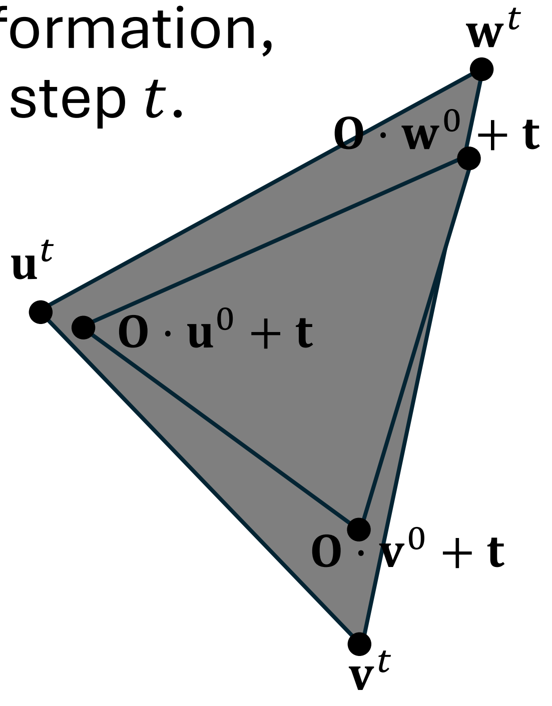
Local:

Given a triangle $\tau = (u, v, w) \in \mathcal{T}$, it has assigned positions from the initial embedding, $\Upsilon^0 : \mathcal{V} \to \mathbb{R}^3$ and from the current embedding $\Upsilon^t : \mathcal{V} \to \mathbb{R}^3$.

We can compute the optimal translation and orthogonal transformation, $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{O} \in \mathbb{R}^{3\times3}$ taking the initial triangle to the triangle at step t.





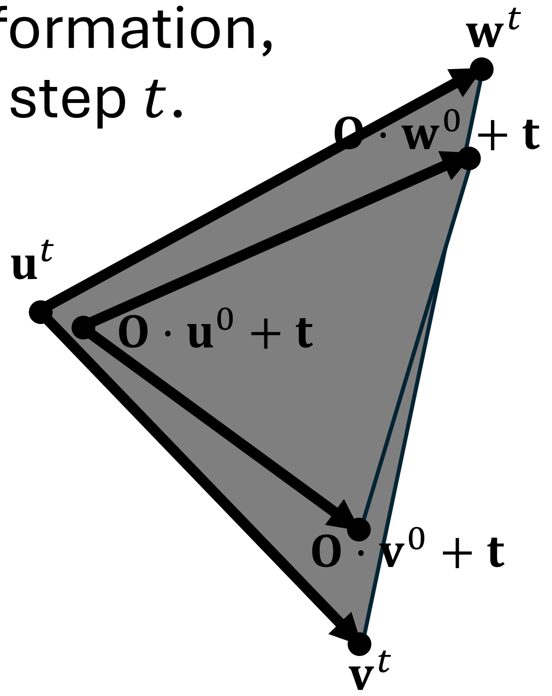


As-Rigid-As-Possible Deformation

Local:

Given a triangle $\tau = (u, v, w) \in \mathcal{T}$, it has assigned positions from the initial embedding, $\Upsilon^0 : \mathcal{V} \to \mathbb{R}^3$ and from the current embedding $\Upsilon^t : \mathcal{V} \to \mathbb{R}^3$.

We can compute the optimal translation and orthogonal transformation, $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{O} \in \mathbb{R}^{3\times 3}$ taking the initial triangle to the triangle at step t.



From u to v, the change in the deformation should be:

$$(\mathbf{O} \cdot \mathbf{v}^0 + \mathbf{t}) - (\mathbf{O} \cdot \mathbf{u}^0 + \mathbf{t}) = \mathbf{O} \cdot \mathbf{v}^0 - \mathbf{O} \cdot \mathbf{u}^0$$

From u to w, the change in the deformation should be:

$$(\mathbf{O} \cdot \mathbf{w}^0 + \mathbf{t}) - (\mathbf{O} \cdot \mathbf{u}^0 + \mathbf{t}) = \mathbf{O} \cdot \mathbf{w}^0 - \mathbf{O} \cdot \mathbf{u}^0$$

As-Rigid-As-Possible Deformation

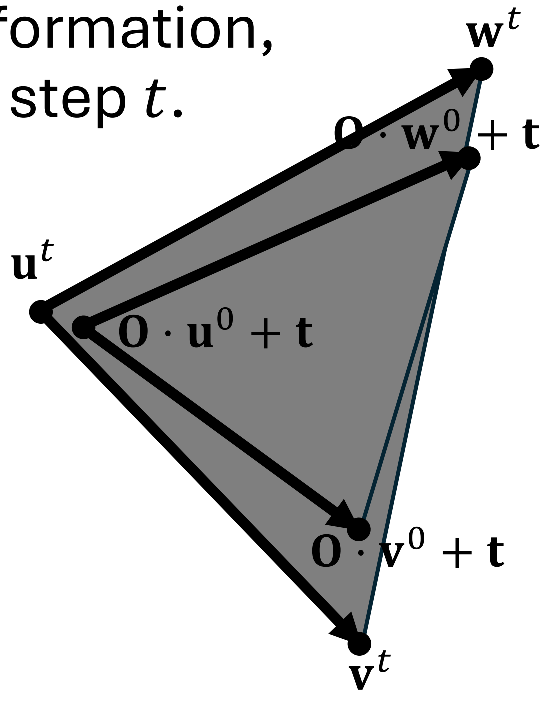
Local:

Given a triangle $\tau = (u, v, w) \in \mathcal{T}$, it has assigned positions from the initial embedding, $\Upsilon^0 : \mathcal{V} \to \mathbb{R}^3$ and from the current embedding $\Upsilon^t : \mathcal{V} \to \mathbb{R}^3$.

We can compute the optimal translation and orthogonal transformation, $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{O} \in \mathbb{R}^{3\times 3}$ taking the initial triangle to the triangle at step t.

$\Rightarrow$ The target differential for the deformation should be:

$$\boldsymbol{\zeta}_{2\tau+0}^{t+1} = \mathbf{O} \cdot \mathbf{v}_0 - \mathbf{O} \cdot \mathbf{u}_0$$
$$\boldsymbol{\zeta}_{2\tau+1}^{t+1} = \mathbf{O} \cdot \mathbf{w}_0 - \mathbf{O} \cdot \mathbf{u}_0$$



As-Rigid-As-Possible Deformation

Global:

Given the target differential $\zeta^{t+1} \in \overline{V}$, we would like to solve the gradient domain problem for the deformation $\Upsilon^{t+1} \in V$ minimizing the energy:

$$
\begin{aligned}
E(\Upsilon^{t+1}) &= \frac{1}{2} \cdot \langle\langle d\Upsilon^{t+1} - \zeta^{t+1}, d\Upsilon^{t+1} - \zeta^{t+1} \rangle\rangle_{\mathcal{M}} \\
&= \frac{1}{2} \cdot \left[\left(d^* \circ \overline{M} \circ d\right)(d\Upsilon^{t+1})\right](d\Upsilon^{t+1}) - \left(d^* \circ \overline{M}\right)(\zeta^{t+1}) + \left[\overline{M}(\zeta^{t+1})\right](\zeta^{t+1}) \\
&= \frac{1}{2} \cdot [S(d\Upsilon^{t+1})](d\Upsilon^{t+1}) - \left(d^* \circ \overline{M}\right)(\zeta^{t+1}) + \cdots
\end{aligned}
$$

... subject to the constraints that, for all $c \in \mathcal{C}$:

$$\Upsilon^{t+1}(c) = \Psi(c)$$

As-Rigid-As-Possible Deformation

Global:

$$E(\Upsilon^{t+1}) = \frac{1}{2} \cdot [S(d\Upsilon^{t+1})](d\Upsilon^{t+1}) - \left(d^* \circ \overline{M}\right)(\zeta^{t+1}) + \cdots$$
$$\Upsilon^{t+1}(c) = \Psi(c)$$

This is a quadratic energy with affine constraints for which the computation of the kernel is simple.

With respect to the scalar field basis $\{\phi_v\}_{v\in\mathcal{V}}$ and the cotangent vector field basis $\{\eta_\tau^1, \eta_\tau^2\}_{\tau\in\mathcal{T}}$, the solution is:

$$\mathbf{v}^{t+1} = \boldsymbol{\iota} \cdot (\boldsymbol{\iota}^\top \cdot \mathbf{S} \cdot \boldsymbol{\iota})^{-1} \cdot \boldsymbol{\iota}^\top \cdot \left(\mathbf{D}^\top \cdot \overline{\mathbf{M}} \cdot \boldsymbol{\zeta}^{t+1} - \mathbf{S} \cdot \hat{\mathbf{v}}\right) + \hat{\mathbf{v}}$$

where $\boldsymbol{\iota}$ is the injection matrix and $\hat{\mathbf{v}}$ are the positions of the locked vertices.

As-Rigid-As-Possible Deformation

Bigger Perspective:

- The processing alternates **local** and **global** computation.
  - Local computation is non-linear, greedily assigning target differentials to individual triangles.
  - Global computation makes the solution consistent by solving a linear system for the vertex positions best-fitting the target differentials.